InfoXtract: A Customizable Intermediate Level Information Extraction Engine[*]

Rohini K. Srihari
Cymfony, Inc.
State University of New York at Buffalo
rohini@Cymfony.com

Wei Li, Cheng Niu and Thomas Cornell
Cymfony Inc.
600 Essjay Road, Williamsville, NY 14221, USA
{wei, cniu, cornell}@Cymfony.com

Keywords: Information Extraction, Named Entity Tagging, Machine Learning, Domain Porting

Abstract

Information extraction (IE) systems assist analysts to assimilate information from electronic documents. This paper focuses on IE tasks designed to support *information discovery* applications. Since information discovery implies examining large volumes of documents drawn from various sources for situations that cannot be anticipated *a priori*, they require IE systems to have breadth as well as depth. This implies the need for a domain-independent IE system that can easily be customized for specific domains: end users must be given tools to customize the system on their own. It also implies the need for defining new *intermediate* level IE tasks that are richer than the subject-verb-object (SVO) triples produced by shallow systems, yet not as complex as the domain-specific scenarios defined by the Message Understanding Conference (MUC). This paper describes a robust, scalable IE engine designed for such purposes. It describes new IE tasks such as entity profiles, and concept-based general events which represent realistic goals in terms of what can be accomplished in the near-term as well as providing useful, actionable information. These new tasks also facilitate the correlation of output from an IE engine with existing structured data. Benchmarking results for the core engine and applications utilizing the engine are presented.

1 Introduction

This paper focuses on new intermediate level information extraction tasks that are defined and implemented in an IE engine, named *InfoXtract*. InfoXtract is a domain independent, but portable information extraction engine that has been designed for information discovery applications.

The last decade has seen great advances in the area of IE. In the US, MUC [Chinchor & Marsh 1998] has been the driving force for developing this technology.

The most successful IE task thus far has been Named Entity (NE) tagging. The state-of-the-art exemplified by systems such as *NetOwl* [Krupka & Hausman 1998], *IdentiFinder* [Miller et al 1998] and InfoXtract [Srihari et al 2000] has reached near human performance, with 90% or above F-measure. On the other hand, the deep level MUC IE task Scenario Template (ST) is designed to extract detailed information for predefined event scenarios of interest. It involves filling the slots of complicated templates. It is generally felt that this task is too ambitious for commercial application at present.

Information Discovery (ID) is a term which has traditionally been used to describe efforts in data mining [Han 1999]. The goal is to extract novel patterns of transactions which may reveal interesting trends. The key assumption is that the data is already in a structured form. ID in this paper is defined within the context of unstructured text documents; it is the ability to extract, normalize/disambiguate, merge and link entities, relationships, and events which provides significant support for ID applications. Furthermore, there is a need to accumulate information across documents about entities and events. Due to rapidly changing events in the real world, what is of no interest one day, may be especially interesting the following day. Thus, information discovery applications demand *breadth* and *depth* in IE technology.

A variety of IE engines, reflecting various goals in terms of extraction as well as architectures are now available. Among these, the most widely used are the *GATE* system from the University of Sheffield [Cunningham et al 2003], the IE components from Clearforest (www.clearforest.com), SIFT from BBN [Miller et al 1998], REES from SRA [Aone & Ramon-Santacruz 1998] and various tools provided by Inxight (www.inxight.com). Of these, the GATE system most closely resembles InfoXtract in terms of its goals as well as the architecture and customization tools. Cymfony differentiates itself by using a hybrid

[*] This work was supported in part by SBIR grants F30602-01-C-0035, F30602-03-C-0156, and F30602-02-C-0057 from the Air Force Research Laboratory (AFRL)/IFEA.

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **2003**	2. REPORT TYPE	3. DATES COVERED **00-00-2003 to 00-00-2003**

4. TITLE AND SUBTITLE **InfoXtract: A Customizable Intermediate Level Information Extraction Enngine**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Cymfony Inc,600 Essjay Road,Williamsville,NY,14221**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**8**	

model that efficiently combines statistical and grammar-based approaches, as well as by using an internal data structure known as a *token-list* that can represent hierarchical linguistic structures and IE results for multiple modules to work on.

The research presented here focuses on a new intermediate level of information extraction which supports information discovery. Specifically, it defines new IE tasks such as *Entity Profile (EP) extraction*, which is designed to accumulate interesting information about an entity across documents as well as within a discourse. Furthermore, *Concept-based General Event (CGE)* is defined as a domain-independent, representation of event information but more feasible than MUC ST.

InfoXtract represents a hybrid model for extracting both shallow and intermediate level IE: it exploits both statistical and grammar-based paradigms. A key feature is the ability to rapidly *customize* the IE engine for a specific domain and application. Information discovery applications are required to process an enormous volume of documents, and hence any IE engine must be able to scale up in terms of processing speed and robustness; the design and architecture of InfoXtract reflect this need.

In the remaining text, Section 2 defines the new intermediate level IE tasks. Section 3 presents extensions to InfoXtract to support cross-document IE. Section 4 presents the hybrid technology. Section 5 delves into the engineering architecture and implementation of InfoXtract. Section 6 discusses domain porting. Section 7 presents two applications which have exploited InfoXtract, and finally, Section 8 summarizes the research contributions.

2 InfoXtract: Defining New IE Tasks

InfoXtract [Li & Srihari 2003, Srihari et al 2000] is a domain-independent and domain-portable, intermediate level IE engine. Figure 1 illustrates the overall architecture of the engine.

A description of the increasingly sophisticated IE outputs from the InfoXtract engine is given below:

- **NE:** Named Entity objects represent key items such as proper names of *person, organization, product, location, target*, contact information such as *address, email, phone number, URL,* time and numerical expressions such as *date, year* and various measurements *weight*, *money*, *percentage*, etc.
- **CE:** Correlated Entity objects capture relationship mentions between entities such as the *affiliation* relationship between a person and his employer. The results will be consolidated into the information object Entity Profile (EP) based on co-reference and alias support.
- **EP:** Entity Profiles are complex rich information objects that collect entity-centric information, in particular, all the CE relationships that a given entity is involved in and all the events this entity is involved in. This is achieved through document-internal fusion and cross-document fusion of related information based on support from co-reference, including alias association. Work is in progress to enhance the fusion by correlating the extracted information with information in a user-provided existing database.
- **GE:** General Events are verb-centric information objects representing 'who did what to whom when and where' at the logical level. Concept-based GE (CGE) further requires that participants of events be filled by EPs instead of NEs and that other values of the GE slots (the action, time and location) be disambiguated and normalized.
- **PE:** Predefined Events are domain specific or user-defined events of a specific event type, such as Product Launch and Company Acquisition in the business domain. They represent a simplified version of MUC ST. InfoXtract provides a toolkit that allows users to define and write their own PEs based on automatically generated PE rule templates.

The InfoXtract engine has been deployed both internally to support Cymfony's *Brand Dashboard*™ product and externally to a third-party integrator for building IE applications in the intelligence domain.

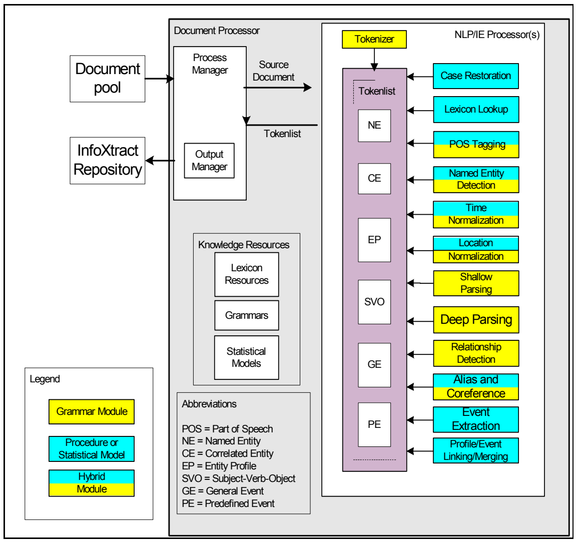


Figure 1: InfoXtract Engine Architecture

3 Hybrid Technology

InfoXtract represents a hybrid model for IE since it combines both grammar formalisms as well as machine learning. Achieving the right balance of these two paradigms is a major design objective of InfoXtract. The core of the parsing and information extraction process in InfoXtract is organized very simply as a pipeline of processing modules. All modules operate on a single in-memory data structure, called a *token list*. A token list is essentially a sequence of tree structures, overlaid with a graph whose edges define relations that may be either grammatical or informational in nature. The nodes of these trees are called tokens. InfoXtract's typical mode of processing is to skim along the roots of the trees in the token list, building up structure "strip-wise". So even non-terminal nodes behave, in the typical case, as complex tokens. Representing a marked up text using trees explicitly, rather than implicitly as an interpretation of paired bracket symbols, has several advantages. For example, it allows a somewhat richer organization of the information contained "between the brackets," allowing us to construct direct links from a root node to its semantic head, for example.

The processing modules that act on token lists can range from lexical lookup to the application of hand written grammars to statistical analysis based on machine learning all the way to arbitrary procedures written in C++. The configuration of the InfoXtract processing pipeline is controlled by a configuration file, which handles pre-loading required resources as well as ordering the application of modules. Despite the variety of implementation strategies available, InfoXtract Natural Language Processing (NLP) modules are restricted in what they can do to the token list to actions of the following three types :

1. Assertion and erasure of token properties (features, normal forms, etc.)
2. Grouping token sequences into higher level constituent tokens.
3. Linking token pairs with a relational link.

Grammatical analysis of the input text makes use of a combination of phrase structure and relational approaches to grammar. Basically, early modules build up structure to a certain level (including relatively simple noun phrases, verb groups and prepositional phrases), after which further grammatical structure is represented by asserting relational links between tokens. This mix of phrase structural and relational approaches is very similar to the approach of Lexical Functional Grammar (LFG) [Kaplan & Bresnan 1982], much scaled down.

Our grammars are written in a formalism developed for our own use, and also in a modified formalism developed for outside users, based on our in-house experiences. In both cases, the formalism mixes regular expressions with boolean expressions. Actions affecting the token list are implemented as side effects of pattern matching. So although our processing modules are in the technical sense token list transducers, they do not resemble Finite State Transducers (FSTs) so much as the regular expression based pattern-action rules used in Awk or Lex. Grammars can contain (non-recursive) macros, with parameters.

This means that some long-distance dependencies, which are very awkward to represent directly in finite state automata can be represented very compactly in macro form. While this has the advantage of decreasing grammar sizes, it does increase the size of the resulting automata. Grammars are compiled to a special type of finite state automata. These token list automata can be thought of as an extension of *tree walking automata* [Mönnich et al 2001, Aho & Ullman 1971, Engelfriet et al 1999]. These are linear automata (as opposed to standard finite state tree automata [Gécseg & Steinby 1997], which are more naturally thought of as parallel) which run over trees. The problem with linear automata on trees is that there can be a number of "next" nodes to move the read head to: *right sister, left sister, parent, first child*, etc. So the vocabulary of the automaton is increased to include not only symbols that might appear in the text (test instructions) but also symbols that indicate where to move the read head (directive instructions). We have extended the basic tree walking formalism in several directions. First we extend the power of test instructions to allow them to check features of the current node and to perform string matching against the semantic head of the current node (so that a syntactically complex constituent can be matched against a single word). Second, we include symbols for action instructions, to implement side effects. Finally, we allow movement not only along the root sequence (string-automaton style) and branches of a tree (tree-walking style) but also along the the terminal frontier of the tree and along relational links.

These extensions to standard tree walking automata extend the power of that formalism tremendously, and could pose problems. However, the grammar formalisms that compile into these token list walking automata are restrictive, in the sense that there exist many token list transductions that are implementable as automata that are not implementable as grammars. Also the nature of the shallow parsing task itself is such that we only need to dip into the reserves of power that this representation affords us on relatively rare occasions. As a result, the automata that we actually plug into the InfoXtract NLP pipeline generally run very fast.

Recently, we have developed an extended finite state formalism named *Expert Lexicon*, following the general trend of lexicalist approaches to NLP. An expert lexicon rule consists of both grammatical components as well as proximity-based keyword matching. All Expert Lexicon entries are indexed, similar to the case for the finite state tool in INTEX [Silberztein 2000]. The pattern matching time is therefore reduced dramatically compared to a sequential finite state device.

Some unique features of this formalism include: (i) the flexibility of inserting any number of Expert Lexicons at any level of the process; (ii) the capability of proximity checking within a window size as rule constraints in addition to pattern matching using an FST call, so that the rule writer can exploit the combined advantages of both; and (iii) support for the propagation of semantic tagging results, to accommodate principles like *one sense per discourse*. Expert lexicons are used in customization of lexicons, named entity glossaries, and alias lists, as well as concept tagging.

Both supervised machine learning and unsupervised learning are used in InfoXtract. Supervised learning is used in hybrid modules such as NE [Srihari et al 2000], NE Normalization [Li et al 2002] and Co-reference. It is also used in the preprocessing module for orthographic case restoration of case insensitive input [Niu et al 2003]. Unsupervised learning involves acquisition of lexical knowledge and rules from a raw corpus. The former includes word clustering, automatic name glossary acquisition and thesaurus construction. The latter involves bootstrapped learning of NE and CE rules, similar to the techniques used in [Riloff 1996]. The results of unsupervised learning can be post-edited and added as additional resources for InfoXtract processing.

Table 1: SVO/CE Benchmarking

	SVO	**CE**
CORRECT	196	48
INCORRECT	13	0
SPURIOUS	10	2
MISSING	31	10
PRECISION	89.50%	96.0%
RECALL	81.67%	82.8%
F-MEASURE	**85.41%**	**88.9%**

Accuracy

InfoXtract has been benchmarked using the MUC-7 data sets which are recognized as standards by the research community. Precision and recall figures for the *person* and *location* entity types were above 90%. For *organization* entity types, precision and recall were in the high 80's reflecting the fact that organization names tend to be very domain specific. InfoXtract provides the ability to create customized named entity glossaries, which will boost the performance of organization tagging for a given domain. No such customization was done in the testing just described. The accuracy of shallow parsing is well over 90% reflecting very high performance part-of-speech tagging and named entity tagging. Table 1 shows the benchmarks for CE relationships which are the basis for EPs and for the SVO parsing which supports event extraction.

4 Engineering Architecture

The InfoXtract engine has been developed as a modular, distributed application and is capable of processing up to 20 MB per hour on a single processor. The system has been tested on very large (> 1 million) document collections. The architecture facilitates the incorporation of the engine into external applications requiring an IE subsystem. Requests to process documents can be submitted through a web interface, or via FTP. The results of processing a document can be returned in XML. Since various tools are available to automatically populate databases based on XML data models, the results are easily usable in web-enabled database applications. Configuration files enable the system to be used with different lexical/statistical/grammar resources, as well as with subsets of the available IE modules.

InfoXtract supports two modes of operation, active and passive. It can act as an active retriever of documents to process or act as a passive receiver of documents to process. When in active mode, InfoXtract is capable of retrieving documents via HTTP, FTP, or local file system. When in passive mode, InfoXtract is capable of accepting documents via HTTP. Figure 2 illustrates a multiple processor configuration of InfoXtract focusing on the typical deployment of InfoXtract within an application.

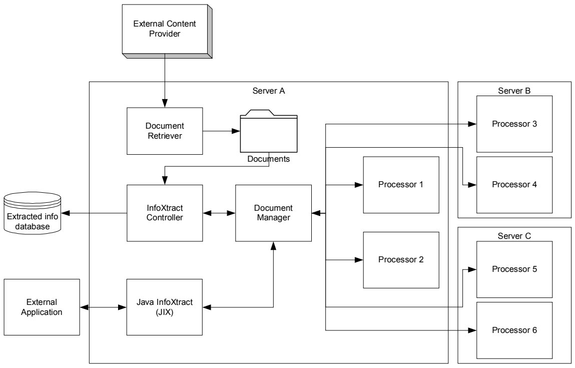


Figure 2: High Level Architecture

The architecture facilitates scalability by supporting multiple, independent Processors. The Processors can be running on a single server (if multiple CPUs are available) and on multiple servers. The Document Manager distributes requests to process documents to all available Processors. Each component is an independent application. All direct inter-module communication is accomplished using the Common Object Request Broker Architecture (CORBA). CORBA provides a robust, programming language independent, and platform neutral mechanism for developing and deploying distributed applications. Processors can be added and removed without stopping the InfoXTract engine. All modules are self-registering and will announce their presence to other modules once they have completed initialization.

The Document Retriever module is only used in the active retriever mode. It is responsible for retrieving documents from a content provider and storing the documents for use by the InfoXtract Controller. The Document Retriever handles all interfacing with the content provider's retrieval process, including interface protocol (authentication, retrieve requests, etc.), throughput management, and document packaging. It is tested to be able to retrieve documents from content providers such as Northern Light, Factiva, and LexisNexis. Since the Document Retriever and the InfoXtract Controller do not communicate directly, it is possible to run the Document Retriever standalone and process all retrieved documents in a batch mode at a later time.

The InfoXtract Controller module is used only in the active retriever mode. It is responsible for retrieving documents to be processed, submitting documents for processing, storing extracted information, and system logging. The InfoXtract Controller is a multi-threaded application that is capable of submitting multiple simultaneous requests to the Document Manager. As processing results are returned, they are stored to a repository or database, an XML file, or both.

The Document Manager module is responsible for managing document submission to available Processors. As Processors are initialized, they register with the Document Manager. The Document Manager uses a round robin scheduling algorithm for sending documents to available Processors. A document queue is maintained with a size of four documents per Processor. The Processor module forms the core of the IE engine. InfoXtract utilizes a multi-level approach to NLP. Each level utilizes the results of the previous levels in order to achieve more sophisticated parsing. The JIX module is a web application that is responsible for accepting requests for documents to be processed. This module is only used in the passive mode. The document requests are received via the HTTP Post request. Processing results are returned in XML format via the HTTP Post response.

In Table 2 we present an example of the performance that can be expected based on the application of all modules within the engine. It should be noted that considerably faster processing per processor can be achieved if output is restricted to a certain IE level, such as named entity tagging only. The output in this benchmark includes all major tasks such as NE, EP, parsing and event extraction as well as XML generation.

This configuration provides throughput of approximately 12,000 documents (avg. 10KB) per day. A smaller average document size will increase the document throughput. Increased throughput can be achieved by dedicating a CPU for each running Processor. Each Processor instance requires approximately 500 MB of RAM to run efficiently. Processing speed increases linearly with additional Processors/CPUs, and CPU speed. In the current state, with no speed optimization, using a bank of eight processors, it is able to process approximately 100,000 documents per day. Thus, InfoXtract is suitable for high volume deployments. The use of CORBA provides seamless inter-process and over-the-wire communication between modules. Computing resources can be dynamically assigned to handle increases in document volume.

Table 2: Benchmark for Efficiency

Server Configuration	2 CPU @ 1 GHz, 2 GB RAM
Operating System	Redhat Linux 7.2
Document Collection Size	500 Documents, 5 MB total size
Engine Configuration	InfoXtract Controller, Document Manager, and 2 Processors running on a single server
Processing Time	30 Minutes

A standard document input model is used to develop effective preprocessing capabilities. Preprocessing adapts the engine to the source by presenting metadata, zoning information in a standardized format and performing restoration tasks (e.g. case restoration). Efforts are underway to configure the engine such that zone-specific processing controls are enabled. For example, zones identified as titles or subtitles must be tagged using different criteria than running text. The engine has been deployed on a variety of input formats including HUMINT documents (all uppercase), the Foreign Broadcast Information Services feed (FBIS), live feeds from content providers such as Factiva (Dow Jones/Reuters), LexisNexis, as well as web pages. A user-trainable, high-performance case restoration module [Niu et al 2003] has been developed that transforms case insensitive input such as speech transcripts into mixed-case before being processed by the engine. The case restoration module eliminates the need for separate IE engines for case-insensitive and case-sensitive documents; this is easier and more cost effective to maintain.

5 Corpus-level IE

Efforts have extended IE from the document level to the corpus level. Although most IE systems perform corpus-level information consolidation at an application level, it is felt that much can be gained by doing this as an extended step in the IE engine. A *repository* has been developed for InfoXtract that is able to hold the results of processing an entire corpus. A proprietary indexing scheme for indexing token-list data has been developed that enables querying over both the linguistic structures as well as statistical similarity queries (e.g., the similarity between two documents or two entity profiles). The repository is used by a *fusion* module in order to generate cross-document entity profiles as well as for text mining operations. The results of the repository module can be subsequently fed into a relational database to support applications. This has the advantage of filtering much of the noise from the engine level and doing sophisticated information consolidation before populating a relational database. The architecture of these subsequent stages is shown in Figure 3.

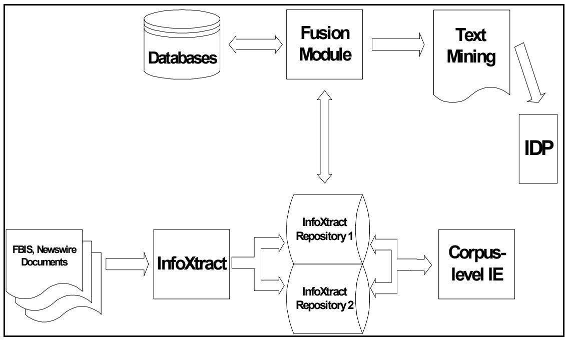


Figure 3: Extensions to InfoXtract

Information Extraction has two anchor points: (i) entity-centric information which leads to an EP, and (ii) action-centric information which leads to an event scenario. Compared with the consolidation of extracted events into cross-document event scenario, cross-document EP merging and consolidation is a more tangible task, based mainly on resolving aliases. Even with modest recall, the corpus-level EP demonstrates tremendous value in collecting information about an entity. This is as shown in Table 3 for only part of the profile of 'Mohamed Atta' from one experiment based on a collection of news articles. The extracted EP centralizes a significant amount of valuable information about this terrorist.

6 Domain Porting

Considerable efforts have been made to keep the core engine as domain independent as possible; domain specialization or tuning happens with minimum change to the core engine, assisted by automatic or semi-automatic domain porting tools we have developed.

Cymfony has taken several distinct approaches in achieving domain portability: (i) the use of a standard document input model, pre-processors and configuration scripts in order to tailor input and output formats for a given application, (ii) the use of tools in order to customize lexicons and grammars, and (iii) unsupervised machine learning techniques for learning new named entities (e.g. weapons) and relationships based on sample seeds provided by a user.

Table 3: Sample Entity Profile

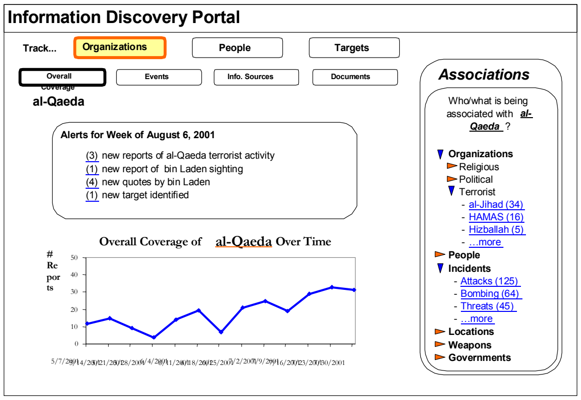
Name	Mohamed Atta
Aliases	Atta; Mohamed
Position	apparent mastermind; ring leader; engineer; leader
Age	33; 29; 33-year-old; 34-year-old
Where-from	United Arab Emirates; Spain; Hamburg; Egyptian; ……
Modifiers	on the first plane; evasive; ready; in Spain; in seat 8D…
Descriptors	hijacker; al-Amir; purported ringleader; a square-jawed 33-year-old pilot; ……
Association	bin Laden; Abdulaziz Alomari; Hani Hanjour; Madrid; American Media Inc.; ……
Involved-events	move-events (2); accuse-events (9), convict-events (10), confess-events (2), arrest-events (3), rent-events (3), ...

It has been one of Cymfony's primary objectives to facilitate domain portability [Srihari 1998] [Li & Srihari 2000a,b, 2003]. This has resulted in a development/customization environment known as the Lexicon Grammar Development Environment (LGDE). The LGDE permits users to modify named entity glossaries, alias lexicons and general-purpose lexicons. It also supports example-based grammar writing; users can find events of interest in sample documents, process these through InfoXtract and modify the constraints in the automatically generated rule templates for event detection. With some basic training, users can easily use the LGDE to customize InfoXtract for their applications. This facilitates customization of the system in user applications where access to the input data to InfoXtract is restricted.

7 Applications

The InfoXtract engine has been used in two applications, the Information Discovery Portal (IDP) and Brand Dashboard (www.branddashboard.com). The IDP supports both the traditional top-down methods of browsing through large volumes of information as well as novel, data-driven browsing. A sample user interface is shown in Figure 4.

Users may select "watch lists" of entities (people, organizations, targets, etc.) that they are interested in monitoring. Users may also customize the sources of information they are interested in processing. Top-down methods include topic-centric browsing whereby documents are classified by topics of interest. IE-based browsing techniques include entity-centric and event-centric browsing. Entity-centric browsing permits users to track key entities (people, organizations, targets) of interest and monitor information pertaining to them. Event-centric browsing focuses on significant actions including money movement and people movement events. Visualization of extracted information is a key component of the IDP. The Information Mesh enables a user to visualize an entity, its attributes and its relation to other entities and events. Starting from an entity (or event), relationship chains can be traversed to explore related items. Timelines facilitate visualization of information in the temporal axis.



Information Discovery Portal

al-Qaeda

Alerts for Week of August 6, 2001

(3) new reports of al-Qaeda terrorist activity
(1) new report of bin Laden sighting
(4) new quotes by bin Laden
(1) new target identified

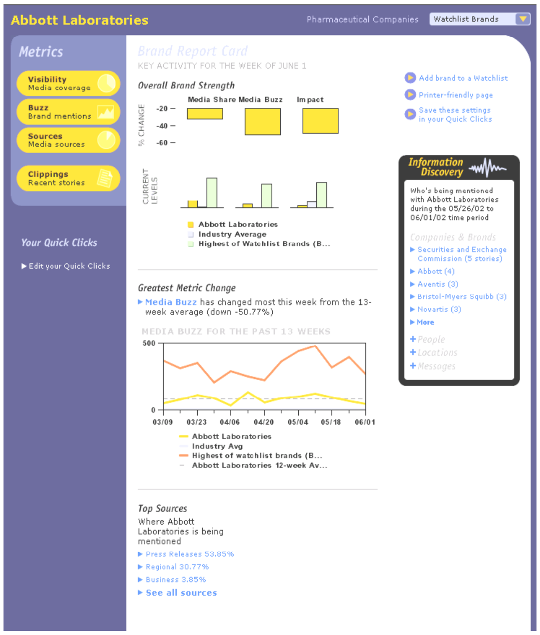
Associations

Who/what is being associated with *al-Qaeda* ?

Figure 4: Information Discovery Portal

Recent efforts have included a tight integration of InfoXtract with visualization tools such as the Web-based Timeline Analysis System (WebTAS) (http://www.webtas.com). The IDP reflects the ability for users to select events of interest and automatically export them to WebTAS for visualization. Efforts are underway to integrate higher-level event scenario analysis tools such as the Terrorist Modus Operandi Detection System (TMODS) (www.21technologies .com) into the IDP.

Brand Dashboard is a commercial application for marketing and public relations organizations to measure and assess media perception of consumer brands. The InfoXtract engine is used to analyze several thousand electronic sources of information provided by various content aggregators (Factiva, LexisNexis, etc.). The engine is focused on tagging and generating brand profiles that also capture salient information such as the descriptive phrases used in describing brands (e.g. *cost-saving, non-habit forming*) as well as user-configurable specific messages that companies are trying to promote and track (*safe and reliable, industry leader*, etc.). The output from the engine is fed into a database-driven web application which then produces report cards for brands containing quantitative metrics pertaining to brand perception, as well as qualitative information describing characteristics. A sample screenshot from Brand Dashboard is presented in Figure 5. It depicts a report card for a particular brand, highlighting brand strength as well as highlighting metrics that have changed the most in the last time period. The "buzz box" on the right hand side illustrates companies/brands, people, analysts, and messages most frequently associated with the brand in question.



Abbott Laboratories

Brand Report Card

KEY ACTIVITY FOR THE WEEK OF JUNE 1

Greatest Metric Change

▶ Media Buzz has changed most this week from the 13-week average (down -50.77%)

Top Sources

Where Abbott Laboratories is being mentioned

▶ Press Releases 53.85%
▶ Regional 30.77%
▶ Business 3.85%
▶ See all sources

Information Discovery

Who's being mentioned with Abbott Laboratories during the 05/26/02 to 06/01/02 time period

Companies & Brands
▶ Securities and Exchange Commission (5 stories)
▶ Abbott (4)
▶ Aventis (3)
▶ Bristol-Myers Squibb (3)
▶ Novartis (3)
▶ More

Figure 5: Report Card from Brand Dashboard

8 Summary and Future Work

This paper has described the motivation behind InfoXtract, a domain independent, portable, intermediate-level IE engine. It has also discussed the architecture of the engine, both from an algorithmic perspective and software engineering perspective. Current efforts to improve InfoXtract include the following: support for more diverse input formats, more use of metadata in the extraction tasks, support for structured data, and capabilities for processing foreign languages. Finally, support for more intuitive domain customization tools, especially the semi-automatic learning tools is a major focus.

Acknowledgments

The authors wish to thank Carrie Pine of AFRL for reviewing and supporting this work.

References

[Aho & Ullman 1971] Alfred V. Aho and Jeffrey D. Ullman. Translations on a context-free grammar. *Information and Control*, 19(5):439–475, 1971.

[Aone & Ramos-Santacruz 1998] REES: A Large-Scale Relation and Event Extraction System. url: http://acl.ldc.upenn.edu/A/A00/A00-1011.pdf

[Chinchor & Marsh 1998] Chinchor, N. & Marsh, E. 1998. MUC-7 Information Extraction Task Definition (version 5.1), *Proceedings of MUC-7.*

[Cunningham et al 2003] Hamish Cunningham et al. Developing Language Processing Components with GATE: A User Guide. http://gate.ac.uk/sale/tao/index.html#annie

[Engelfriet et al 1999] Joost Engelfriet, Hendrik Jan Hoogeboom, and Jan-Pascal Van Best. Trips on trees. *Acta Cybernetica*, 14(1):51–64, 1999.

[Gécseg & Steinby 1997] Ferenc Gécseg and Magnus Steinby. Tree languages. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages: Beyond Words*, volume 3, pages 1–68, Berlin, 1997. Springer

[Han 1999] Han, J. Data Mining. 1999. In J. Urban and P. Dasgupta (eds.), Encyclopedia of Distributed Computing, Kluwer Academic Publishers.

[Hobbs 1993] J. R. Hobbs, 1993. FASTUS: A System for Extracting Information from Text, Proceedings of the DARPA workshop on Human Language Technology", Princeton, NJ, 133-137.

[Kaplan & Bresnan 1982] Ronald M. Kaplan and Joan Bresnan. Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–281. The MIT Press, Cambridge, MA, 1982.

[Krupka & Hausman 1998] G. R Krupka and K. Hausman, "IsoQuest Inc: Description of the NetOwl Text Extraction System as used for MUC-7", MUC-7

[Li et al 2002] Li, H., R. Srihari, C. Niu, and W. Li (2002). Localization Normalization for Information Extraction. COLING 2002, 549–555, Taipei, Taiwan.

[Li, W & R. Srihari 2000a]. A Domain Independent Event Extraction Toolkit, Final Technical Report, Air Force Research Laboratory, Information Directorate, Rome Research Site, New York

[Li, W & R. Srihari 2000b]. Flexible Information Extraction Learning Algorithm, Final Technical Report, Air Force Research Laboratory, Information Directorate, Rome Research Site, New York

[Li & Srihari 2003] Li, W. and R. K. Srihari (2003) Intermediate-Level Event Extraction for Temporal and Spatial Analysis and Visualization, Final Technical Report AFRL-IF-RS-TR-2002-245, Air Force Research Laboratory, Information Directorate, Rome Research Site, New York.

[Miller et al 1998] Miller, Scott; Crystal, Michael; Fox, Heidi; Ramshaw, Lance; Schwartz, Richard; Stone, Rebecca; Weischedel, Ralph; and Annotation Group, the 1998. *Algorithms that Learn to Extract Information; BBN: Description of the SIFT System as Used for MUC*-7.

[Mönnich et al 2001] Uwe Mönnich, Frank Morawietz, and Stephan Kepser. A regular query for context-sensitive relations. In Steven Bird, Peter Buneman, and Mark Liberman, editors, *IRCS Workshop Linguistic Databases 2001*, pages 187–195, 2001

[Niu et al 2003] Niu, C., W. Li, J. Ding, and R.K. Srihari (to appear 2003). Orthographic Case Restoration Using Supervised Learning Without Manual Annotation. *Proceedings of The 16th FLAIRS*, St. Augustine, FL

[Riloff 1996] [Automatically Generating Extraction Patterns from Untagged Text. AAAI-96.

[Roche & Schabes 1997] Emmanuel Roche & Yves Schabes, 1997. Finite-State Language Processing, The MIT Press, Cambridge, MA.

[Silberztein 1999] Max Silberztein, (1999). *INTEX: a Finite State Transducer toolbox*, in Theoretical Computer Science #231:1, Elsevier Science

[Srihari 1998]. A Domain Independent Event Extraction Toolkit, AFRL-IF-RS-TR-1998-152 Final Technical Report, Air Force Research Laboratory, Information Directorate, Rome Research Site, New York

[Srihari et al 2000] Srihari, R, C. Niu and W. Li. (2000). A Hybrid Approach for Named Entity and Sub-Type Tagging. In *Proceedings of ANLP 2000*, 247–254, Seattle, WA.